Multipath Bandwidth Scavenging in the Internet of Things

Isabel Montes[1(✉)], Romel Parmis[1], Roel Ocampo[1], and Cedric Festin[2]

[1] Computer Networks Laboratory, Electrical and Electronics Engineering Institute, University of the Philippines Diliman, Quezon City, Philippines
isabel.montes@upd.edu.ph

[2] Networks and Distributed Systems Group, Department of Computer Science, University of the Philippines Diliman, Quezon City, Philippines

Abstract. In order to achieve the capacity and geographic scope required by IoT applications, service providers should explore mutually-beneficial modes of collaboration such as through cooperative packet forwarding by IoT nodes and cooperative gatewaying through fixed backhauls. To promote such resource pooling while minimizing negative impact on collaborating providers, we developed a transport-layer approach that combines multipath techniques with less-than-best effort (LBE) congestion control methods to enable IoT nodes to opportunistically scavenge for idle bandwidth across multiple paths. Initial tests using TCP-LP and LEDBAT congestion control algorithms on scavenging secondary flows show that this desired functionality can be achieved, while our use of standard TCP congestion control on primary flows ensures that IoT nodes are guaranteed at least one flow that can compete for fair share of the network capacity.

Keywords: Internet of things · Bandwidth scavenging · Less-than-best-effort · Congestion control · Multipath flows

1 Introduction

The Internet of Things (IoT) will place new demands on service providers for connectivity, including fixed infrastructure-based gateway services and backhaul to cloud services for further aggregation, processing, storage and distribution of data from devices and smart objects. Prudent design would dictate that such gateways and backhauls be engineered to appropriately handle peak aggregate traffic from potentially large numbers of data sources. In addition, sufficient coverage over large geographic areas would ensure continuous connectivity even in the face of mobility. Moreover, the strategic placement of access points and gateways would help minimize energy-consuming packet forwarding within the wireless network of objects.

All these technical requirements will result in design challenges and significant capital and operating costs to IoT service providers in the future. In order to avoid having to engineer for peak loads and maximum coverage, one possible

R. Giaffreda et al. (Eds.): IoT360 2014, Part I, LNICST 150, pp. 297–304, 2015.
DOI: 10.1007/978-3-319-19656-5_42

approach for IoT providers servicing overlapping areas would be to enter into mutually-beneficial bilateral commercial agreements enabling cooperative access and transit through their peers nodes and infrastructure. Such cooperation may further be enhanced if nodes are available to concurrently exploit the multiple forwarding paths through the additional resources made available by cooperating providers.

1.1 Multipath Bandwidth Scavenging

Although current routing techniques allow packets from a single origin to be forwarded via multiple routes and gateways, naively striping packets from a single flow into multiple paths may cause problems for transport layer protocols with reliable in-order delivery and congestion control functionality. Uneven path delays and loss characteristics may trigger timeouts and unnecessary retransmissions, requiring endpoints to heavily buffer out-of-order received packets [4,5,16]. A better alternative would be to intelligently partition application flows into subflows and enforce per-subflow reliability and congestion control mechanisms. This has been the general approach taken by the Internet community with Multipath TCP (MPTCP), which, along with coordinated congestion control between subflows, will provide TCP the capability to utilize multiple paths between source and destination for redundancy and better resource usage [15].

MPTCP can potentially provide the multipath capability we require. However, it is possible that a provider will not wish to fairly share bandwidth with a competitor that opportunistically uses bandwidth on top of what it can already obtain from its own network. In other words, an IoT service provider might only allow a competitor to scavenge whatever remaining unused bandwidth is available.

1.2 Less-Than-Best Effort Congestion Control as a Scavenging Mechanism

If opportunistic scavenging subflows are too aggressive, these may negatively impact the ability of other nodes to use the network and thus defeat the purpose of allowing scavenging in the first place. Indeed, a paramount concern in scavenging scenarios is to minimize impact on entities volunteering the use of their idle resources [13]. This may be achieved through the use of a class of congestion control mechanisms called less-than-best-effort (LBE) mechanisms, that detect the onset of congestion more quickly than conventional packet loss-based ones [14]. A desirable side effect is that when LBE flows mix with TCP flows in a bottleneck link, the former yield bandwidth to the latter. In the absence of any competing flows, an LBE flow will also attempt to maximize the use of the available bandwidth. These characteristics make LBE congestion control techniques well suited to the task of opportunistic bandwidth scavenging.

Inspired by current work on LBE congestion control and MPTCP, we developed a hybrid transport-layer approach that would enable concurrent multipath bandwidth scavenging by combining MPTCPs multipath mechanisms with LBE congestion control. Our design and validation efforts are described in the rest of this paper.

2 MP-LBE Design

Similar to MPTCP, two communicating endpoints start by establishing a single primary subflow that uses standard TCP-like congestion control. Any additional paths that are discovered will carry secondary subflows, and use LBE congestion control mechanisms. These secondary subflows are essentially the ones that scavenge bandwidth from the rest of the network, opportunistically using resources from both its own and from cooperating providers.

2.1 Congestion Control in Secondary Subflows

Our work aims to explore the use of the LBE class of congestion control methods in secondary subflows in order to achieve low-impact multipath bandwidth scavenging. We start by evaluating TCP-LP and LEDBAT as candidate congestion control methods in our secondary subflows.

TCP-LP is a congestion control algorithm that manages the congestion window of the sender based on the delay experienced by the traffic on a bottleneck [9]. TCP-LP uses variations in one-way delay to infer congestion earlier than standard TCP through a simple threshold-based algorithm. TCP-LP calculates one-way delay (*owd*) upon receiving an ACK by computing the difference between the receiver's timestamp in the ACK and the timestamp taken when the sender sent the packet, which the receiver copies into the ACK and echoes back to the sender. A smoothing parameter γ is used to get a weighted average of the current *owd* measurement with the previous *owd* measurement. When the smoothed *owd* rises above a fraction δ of the difference between the maximum *owd* (d_{max}) and minimum *owd* (d_{min}), congestion is inferred. TCP-LP will then reduce the congestion window by half and enter an inference phase where it awaits further congestion indication. If congestion is detected during the inference phase, *cwnd* is reduced to 1. Otherwise, TCP-LP proceeds with an additive increase of *cwnd*.

LEDBAT, like TCP-LP, measures *owd* using the timestamps in the ACKs received at the sender side [12]. In place of TCP-LPs threshold-based algorithm for inferring congestion, LEDBAT makes use of a target queuing delay value. When queuing delay becomes higher than the target, congestion is assumed and LEDBAT reduces its *cwnd* to alleviate the potential congestion in the network. LEDBAT regulates the *cwnd* size using a proportional-integral-derivative (PID) controller, which varies the *cwnd* proportional to the difference of the queueing delay and the target value.

2.2 Congestion Control in Primary Subflows

Our primary subflows use TCP SACK, which is a loss-based congestion control algorithm. As such, it does not detect congestion as early as the delay-based algorithms of TCP-LP and LEDBAT.

3 Evaluation

We used an existing NS-2 MPTCP implementation [11] and disabled congestion control coupling between subflows. We further modified it by assigning its first subflow to use standard TCP, and any succeeding subflow added to the connection takes on an LBE congestion control algorithm. We implemented two versions of MP-LBE: one that uses LEDBAT for its secondary flow and one that uses TCP-LP. Our LEDBAT implementation uses a target queueing delay value of 12 ms. We used γ=1/8 and δ=0.25 for the TCP-LP implementation.

The topology used in all the simulations is shown in Fig. 1. An MP-LBE connection is configured with two subflows, one primary and one secondary subflow, and each of these share a bottleneck link with a TCP connection. The bottleneck links each have a capacity of 5 Mbps and 5 ms delay. The link used by the primary subflow will be referred to as the top link, while the link used by the secondary subflow will be referred to as the bottom link.

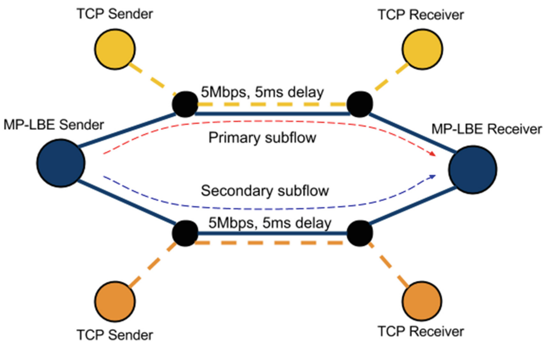
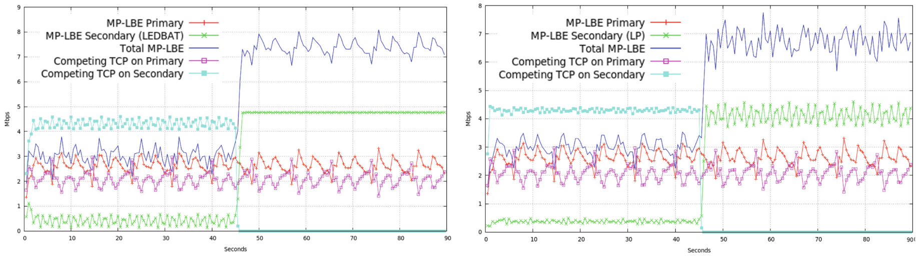


Fig. 1. The topology used for the simulations. Each access link is configured with a capacity of 100 Mbps, with 5 ms delay.

3.1 Bandwidth Scavenging

In this simulation, both subflows compete with standard TCP traffic. MP-LBE's primary subflow should share its evenly link with the competing traffic, while the secondary subflow should back off. Halfway into the simulation, the TCP connection on the bottom link ends, and the secondary subflow should react by maximizing the available bandwidth once the link becomes idle. Both MP-LBE (LEDBAT) and MP-LBE (TCP-LP) are able to achieve this behavior, as seen in Fig. 2. When the secondary flow is using LEDBAT,it is able to maximize the available bandwidth better than TCP-LP. We observed that LEDBAT achieves a steadier throughput because its *cwnd* size does not change as drastically as that of TCP-LP.



(a) LEDBAT (b) TCP-LP

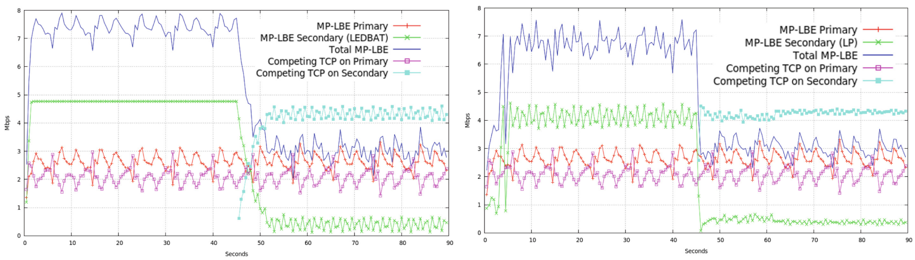
Fig. 2. Bandwidth scavenging behavior of MP-LBE.

3.2 LBE Behavior

In this experiment, we disabled the competing TCP connection on the bottom link at the start of the simulation. This allowed the secondary flow to maximize 5 Mbps capacity of the link. At 45 s, a TCP connection begins on the bottom link, which should cause the secondary subflow to back off. Figure 3 shows the simulation results. MP-LBE using TCP-LP on its secondary subflow was able to back off more rapidly than LEDBAT, but both demonstrated correct LBE behavior when the bottom link stopped being idle.

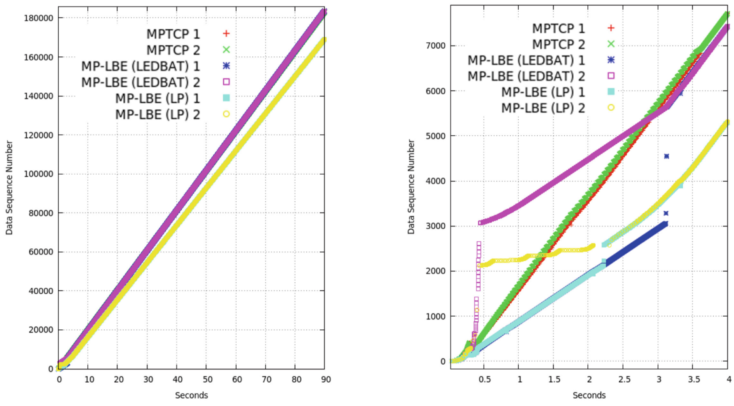
3.3 Goodput

In multipath connections, even if the aggregation of bandwidth effectively improves throughput, the goodput achieved is usually not as high due to the out-of-order arrival of packets. To evaluate the MP-LBEs goodput performance, we recorded the data-level sequence numbers (DSNs) received by the destination node and plotted this against time. For this experiment, we eliminated all competing TCP traffic. We ran the simulation using regular MPTCP, in addition to



(a) LEDBAT (b) TCP-LP

Fig. 3. LBE Behavior of MP-LBE.



(a) (b)

Fig. 4. DSNs received at the destination. The y-axis is scaled to 1:536, as DSNs are in increments of 536.

the simulation runs for MP-LBE (LEDBAT) and MP-LBE (TCP-LP). The simulation results (Fig. 4) show that the rate of DSN increase of both MPTCP and MP-LBE (LEDBAT) is the same, while MP-LBE (TCP-LP) achieves much lower DSN within the given time. For the first 4 s of the experiment (Fig. 4(b)), the MP-LBEs primary subflow (for both LEDBAT and TCP-LP) received DSNs at a slower rate than the secondary subflow, causing the goodput to suffer because only the DSNs on the secondary subflow are arriving, and all the DSNs sent through the primary subflow are delayed in arriving. In the case of MP-LBE (LEDBAT), when the delayed packets finally arrive a little after 3 s, the primary subflow has picked up its pace. After this point, the rate of DSN increase of MP-LBE (LEDBAT) matches that of MPTCP.

4 Related Work

Resource scavenging is not a new concept, having been previously used to harness idle computing resources to perform useful calculations for users other than the resource owner [13]. Our approach focuses on network bandwidth scavenging through a transport layer multipath approach. While there has been some recent similar work on the development of a multipath version of LEDBAT called LEDBAT-MP [1], we are interested in the more general class of LBEs and intend to comparatively evaluate several of the representative algorithms for our intended application. Furthermore, our approach makes a crucial distinction between primary and secondary flows, ensuring that nodes can rely on at least one flow, the primary one, to compete fairly within the network.

In order to achieve cooperative gatewaying among providers, we need mechanisms to enable concurrent access to their respective fixed wireless infrastructure.

BeWifi [2], a service rolled out by service provider Telefonica, allows users to use idle capacity through neighbors access points within range, while CableWiFi [3] employs a multi-provider model, allowing customers from five ISPs access to the consortiums infrastructure. One mechanism that can enable cooperative gatewaying is offered by BaPu (Bunching of Access Point Uplinks) [8], which employs packet overhearing to pool together WiFi uplinks that are in close proximity to one another. BaPu was designed primarily for uploading user-generated content over the Internet and cannot be used for downloads.

The ability to concurrently exploit multiple paths for bandwidth scavenging may also be viewed as a problem of bandwidth aggregation. Application layer solutions such as DBAS [7] typically do not require changes in the underlying protocols or infrastructure, but instead rely on an application-layer proxy residing within the endpoint to intercept traffic, and manage the scheduling, reordering, and transmission of that traffic over multiple network interfaces. We preferred to take an endpoint-based approach since it offers an end-to-end solution, covering both the fixed and wireless portions of the network.

5 Conclusion and Future Work

We proposed an approach for multipath bandwidth scavenging in the Internet of Things through the introduction of a transport-layer protocol that uses TCP-like congestion control for primary subflows and LBE congestion control for secondary subflows. Our simulations demonstrate that our MP-LBE design can effectively improve throughput when an idle link becomes available for a secondary subflow to utilize. Such an ability to scavenge additional bandwidth and paths may offer the ability for sensors and devices to transmit information at higher-than-minimum levels of spatial and temporal resolution, or to explore shortcut fast paths to accelerate local aggregation and processing of data by peer devices within the IoT. When no additional links are available, the primary subflow is able to maintain the throughput of a standard single-flow TCP, and secondary flows are able to rapidly utilize the links once they become idle.

Smart objects and devices in the Internet of Things will undoubtedly have to dedicate most of their resources to their primary tasks of sensing and aggregating data, and to execute any local processing and cognitive functionality required. It is a challenge therefore to minimize the resource footprint of any new functionality being introduced, including multipath bandwidth scavenging. The management of concurrent multipath TCP flows is known to introduce additional buffering resource requirements in endpoints [6], with approaches ranging from smart packet transmission scheduling [4,5] and network coding [10] having been proposed as mitigating solutions. We intend to investigate these approaches further, with the view of achieving low-overhead implementations.

References

1. Adhari, H., Werner, S., Dreibholz, T., Rathgeb, E.P: LEDBAT-MP on the application of lower-than-best-effort for concurrent multipath transfer. In: Proceedings of

the 4th International Workshop on Protocols and Applications with Multi-Homing Support (PAMS), Victoria, British Columbia/Canada, May 2014. ISBN 978-1-4799-2652-7
2. BeWifi, Accessed on 30 June 2014. http://www.bewifi.es
3. Cable WiFi : Internet access brought to consumers through a collaboration among U.S. Internet Service Providers, Accessed on 30 June 2014. http://www.cablewifi.com
4. Chebrolu, K., Raman, B., Rao, R.R.: A network layer approach to enable TCP over multiple interfaces. Wirel. Netw. **11**(5), 637–650 (2005)
5. Evensen, K., Kaspar, D., Engelstad, P., Hansen, A.F., Griwodz, C., Halvorsen, P: A network-layer proxy for bandwidth aggregation and reduction of ip packet reordering. In: IEEE 34th Conference on Local Computer Networks. LCN 2009, pp. 585–592. IEEE (2009)
6. Ford, A., Raiciu, C., Handley, M., Barre, S., Inyegar, J.: Architectural Guidelines for Multipath TCP Development. RFC 6182, RFC Editor, March 2011
7. Habak, K., Youssef, M., Harras, K.A.: DBAS: a deployable bandwidth aggregation system. In: 2012 5th International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–6, May 2012
8. Jin,T., Huu, T.V., Blass, E.-O., Noubir, G.: BaPu: efficient and practical bunching of access point uplinks. CoRR, abs/1301.5928, 2013
9. Kuzmanovic, A., Knightly, E.W.: TCP-LP: low-priority service via end-point congestion control. IEEE/ACM Trans. Netw. **14**(4), 739–752 (2006)
10. Li, M., Lukyanenko, A., Tarkoma, S., Cui, Y., Yl-Jski, A.: Tolerating path heterogeneity in multipath TCP with bounded receive buffers. Comput. Netw. **64**, 1–14 (2014)
11. Google Code Project. Multipath-TCP: Implement Multipath TCP on NS-2, Accessed on 30 June 2014. http://code.google.com/p/multipath-tcp
12. Shalunov, S., Hazel, G., Iyengar, J., Kuehlewind, M.: Low Extra Delay Background Transport (LEDBAT). RFC 6817, RFC Editor, December 2012
13. Strickland, J.W., Freeh, V.W., Ma, X., Vazhkudai, S.S.: Governor: autonomic throttling for aggressive idle resource scavenging. In: Second International Conference on Autonomic Computing, ICAC 2005. Proceedings, pp. 64–75. IEEE (2005)
14. Welzl, M., Ros, D.: A Survey of Lower-than-Best-Effort Transport Protocol. RFC 6297, RFC Editor, June 2011
15. Wischik, D., Raiciu, C., Greenhalgh, A., Handley, M.: Design, implementation and evaluation of congestion control for multipath TCP. NSDI **11**, 8–8 (2011)
16. Zinner, T., Tutschku, K., Nakao, A., Tran-Gia, P.: Using concurrent multipath transmission for transport virtualization: analyzing path selection. In: 2010 22nd International, Teletraffic Congress (ITC), pp. 1–7, Sept 2010